**Trabalho React**

Novos Recursos:

Automatic Batching:

Batching é quando o React agrupa várias atualizações de estado em uma única re-renderização para melhor desempenho.

É ótimo para o desempenho porque evita reprocessamentos desnecessários. Também evita que seu componente renderize estados “inacabados” onde apenas uma variável de estado foi atualizada, o que pode causar bugs.

Transition:

É um novo conceito no React para distinguir entre atualizações urgentes e não urgentes.

Suspense:

Suspense permite que você especifique declarativamente o estado de carregamento de uma parte da árvore de componentes se ainda não estiver pronta para ser exibida.

Suspense no React 18 funciona melhor quando combinado com a API de transição. Se você suspender durante uma transição, o React impedirá que o conteúdo já visível seja substituído por um fallback. Em vez disso, o React atrasará a renderização até que dados suficientes sejam carregados para evitar um estado de carregamento incorreto.

Novos HOOKS:

useld:

O useld é um novo hook para gerar IDs exclusivos no cliente e no servidor, evitando incompatibilidades de hidratação.

Permite que você defina uma função que será executada apenas quando o componente for montado ou atualizado, mas sem bloquear a renderização. Em vez disso, a renderização do componente será adiada até que a função retorne os dados necessários.

useTransition:

O start transiton é o novo conceito para distinguir entre atualizações urgentes e não urgentes.

Deve ser utilizado quando tivermos controle da rendeziração pesada.

useDeferredValue:

O useDeferredValue é bem semelhante ao useTransition porem, permite adiar a nova renderização de uma parte não urgente da árvore.

Deve ser util em situações em que você precisa renderizar um valor em um componente que não está disponível imediatamente, como em casos de carregamento de dados de uma fonte externa, por exemplo, uma API.

useSyncExternalStore :

É um novo hook que permite que os armazenamentos externos suportem leituras simultâneas, forçando as atualizações no armazenamento a serem síncronas.

Deve ser em situações em que você precisa manter o estado de um componente sincronizado com dados em tempo real de uma fonte externa, como em aplicativos de colaboração em tempo real ou jogos online.

useInsertionEffect:

É um novo hook que permite que as bibliotecas CSS-in-JS resolvam problemas de desempenho de injeção de estilos na renderização.

Dever ser util em situações em que você deseja adicionar uma animação para a inserção de novos elementos em uma lista ou conjunto de elementos, como em aplicativos de galeria de imagens ou listas de tarefas.

Projeto Segunda Etapa

Eu vou utilizar a api <https://rickandmortyapi.com/api> e irei fazer o sistema de busca de personagem e adicionar os personagens favorito em uma aba separada.

Eu vou utilizar o hook useTransition para poder ter um melhor controle sobre a renderização pesada do projeto, o useState para armazenar e gerenciar o estado do componente.